Multi-camera complexity assessment system for assembly line work stations

Karel Bauters
Hendrik Van Landeghem
University of Ghent
Department of Industrial Management
Technologiepark 903
B-9052 Zwijnaarde, Belgium
E-mail: karel.bauters@ugent.be

Dirk Van Haerenborgh
Maarten Slembrouck
Dimitri Van Cauwelaert
David Van Hamme
Peter Veelaert
University College Gent
Faculty of applied engineering sciences
Schoonmeersstraat 52
BE-9000 Ghent, Belgium

Wilfried Philips
University of Ghent
TELIN-IPI-IMINDS
Sint Pietersnieuwstraat 41
B-9000 Ghent, Belgium

KEYWORDS

Multi-camera, manufacturing complexity, production engineering, visual hull, image analysis

ABSTRACT

In the last couple of years, the market demands an increasing number of product variants. This leads to an inevitable rise of the complexity in manufacturing systems. A model to quantify the complexity in a workstation has been developed, but part of the analysis is done manually. Thereto, this paper presents the results of an industrial proof-of-concept in which the possibility of automating the complexity analysis using multi-camera video images, was tested.

INTRODUCTION

Manufacturing plants are constantly pushed towards higher quality, lower cost and more product variety. Increased product variety is necessary to meet the changing customer and sustainability demands, but it also entails an increase of the complexity of (re)designing processes and workstations.
Zeltzer et al. (2012) proposed a clear and objective definition of the complexity of a workstation. Furthermore, the main drivers of complexity were determined and used to analyze work stations and categorize them as high complex or low complex systems.
Some of the information needed for this complexity assessment can be captured directly from the Enterprise Resource Planning (ERP) or Manufacturing Execution System (MES), but part of the data capturing is still done manually. This paper presents a field-test within a company in the automotive sector, where multi-camera video footage was used to automate the data capturing at the workstation. The workstation that was investigated is part of the rear axle assembly line, which produces rear axles for 3 different models.
The paper starts with an introduction to the video analysis technology and the explanation of the algorithms used to process the images. Further, we clarify how the captured data was translated in information that is useful for the complexity model and we present some results. We then finalize the paper with the conclusions and future research.

LITERATURE REVIEW

Not a lot of research had been done to date on the level of complexity of a manufacturing system. The first effort to quantify the effect of complexity on the performance of automotive plants was made by MacDuffie et al. Following their research, the part complexity is the only element that has a consistent negative effect on the performance of such a production system. ElMaraghy et al. (2003) state that the complexity is related directly to the quantity, diversity and content of the information that is passed to the human in the system. They also captured the complexity of a production system in an index, which is primarily based on the information needs (ElMaraghy et al, 2004). Zeltzer et al. (2012) proposed a clear and objective definition of the complexity of a workstation. They were also the first to develop a model that quantifies the relationship between complexity as perceived by the operators and its drivers.
The use of video recordings in a manufacturing environment is not completely new. For years, industrial engineers are using films to perform time studies using Predetermined Motion and Time Systems (PMTS). Video clips are very efficient to document the work method (Karger and Hancock 1982, Konz 2001), but the inability to derive exact distances from the images lead to inaccuracies in the results. Elnekave and Gilad (2006) developed a rapid video-based analysis system that is able to translate distances accurately from the picture frame into real distance values of the workstation by using digital mapping. Dencker et al. (1999) presented a video-based system which served as a training tool for operators on the one hand and as an information system on operations on the other hand. Furthermore questions of ergonomics or health and safety issues at the workstation were taken into account. At Nexteer, a supplier of automotive parts, 2D video analysis software is used for continuous improvement of their processes. By overlaying the images of 3 different operators at the same workstation, differences are demonstrated and used to improve and standardize the work method.
Although video images are commonly used in industry, there is to the best of our knowledge no system to date that uses multi-

camera footage to determine the complexity of workstations and detect waste in assembly processes.

VISUAL HULL EXTRACTION

Position extraction

To calculate the position of the subject on the ground plane, a multi-camera setup is utilized. This setup enables us to approximate the 3D shape of an object in the overlapping field of view (FOV) of the cameras, otherwise called the visual hull (Laurentini, 1994). A visual hull is generated by first constructing, for each camera, a generalized infinity cone in the 3D space with the camera position being the apex, and the silhouette in the camera view as the base. The visual hull is then the intersection of these cones. We approximate the position of the subject by projecting the visual hull's center of mass onto the ground plane.

Silhouette extraction

The silhouettes, commonly represented as binary masks, are typically produced by foreground/background segmentation algorithms based on, amongst others, static background models (Kim et al., 2007) or motion detection (Zivkovic et al., 2006). Commonly, such algorithms are only able to provide usable foreground masks when operating in a highly controlled environment. The ambient space in a factory hall however, is largely composed of moving objects. Moreover, many factory halls have large north face windows in the roof, bringing additional global lighting changes. Therefore, typical foreground/background segmentation methods proved to be unsuitable in this application. To overcome these issues, we outfitted the subject with a yellow fluorescent vest that is clearly distinguishable from the background. Hence, the segmentation process is primarily based on color information. We propose to convert the image to YUV color space for enhanced robustness against lighting changes. The process of extracting the foreground from the input video sequences is explained in Figure 1.

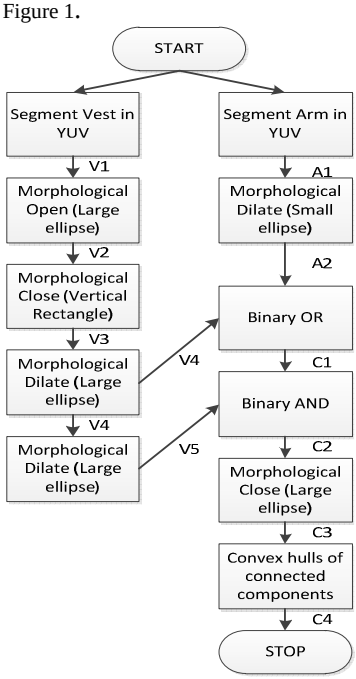

Figure 1: Segmentation algorithm outline

The first stage is to obtain a rough segmentation of the fluorescent vest from the input Figure 2 by using an empirically defined double threshold in the YUV color space. This segmentation (V1) is then post processed using a morphological opening to eliminate noise (V2). Since the fluorescent vest is supplemented with gray horizontal stripes, an additional morphological close operation with a large vertical rectangular structuring element is performed. At this stage, a rough segmentation of this vest is attained (V3). Incidentally, this mask is frequently partially occluded by the bare arms of the subject. To incorporate this, an additional double thresholding step is introduced to obtain a rough segmentation (A1) of the arms of the subject from the input image. Then, V3, and A1 are dilated with an ellipsoidal structuring element, in order to make these masks overlap each other when supplemented on each other (C1). Since bright yellow is far less likely to occur than skin color, a region of interest is generated by dilating V4, the result of which is subject to a binary and operation with C1. Any residual holes in C2 are filled with a morphological closing operation. Finally, we eliminate any concavities in the resulting foreground regions by calculating the convex hull of each of the connected components. This is beneficial in computing the visual hull, as these concavities typically depict the shoulders of the object. Figure 3 shows the resulting foreground mask for a single camera.

Figure 2: input camera image

Figure 3: final foreground mask of single camera

3D shape reconstruction

In this work, the visual hull is constructed by means of voxel carving. First, the 3D space is discretized into voxels. Any voxel is part of the visual hull if its projection onto each of the cameras planes lies inside the respective silhouette. In essence, each camera carves away regions of the 3D shape that do not project onto its silhouette, resulting in a recognizable 3D

shape. Figure 4 shows the generation of the visual hull of a person for 1, 2, 3 and 4 cameras respectively.
Note that to guarantee a precise visual hull, the quality of both camera calibration parameters (both intrinsic and extrinsic) and the extracted silhouettes are paramount. Furthermore, the visual hull is not necessarily convex, since cavities in the silhouettes are carved out of the 3D shape as well. As the process of depth carving solely considers light rays being blocked by the actual object, concave surfaces on the real object are represented as planar surfaces on the visual hull. This problem can be resolved, however partially, by increasing the number of distinct viewing angles as this increases the level of detail of the resulting 3D object, or by changing the position of the cameras so as to view the concavity from a more sideways position. Indeed, the visual hull is highly dependent on the camera positions.

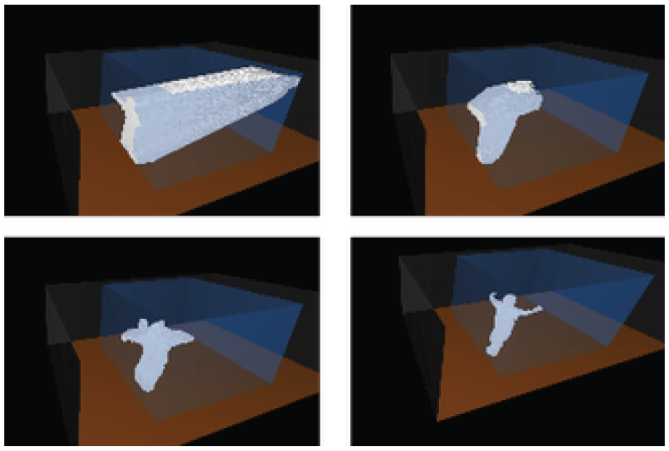

Figure 4: Principle of voxel carving with 4 viewpoints

DATA PROCESSING

The output of the image processing is a sequential list of 3-dimensional coordinates which describe the position of the operator for every frame (20ms) in the video recordings. Theoretically, three cameras should be sufficient to calculate these positions. In practice, adding a fourth camera helps to eliminate noise in these results. This is shown in Figure 5 and Figure 6 in which all positions visited by the operator during the recordings are depicted, based on the data obtained from respectively three and four cameras.

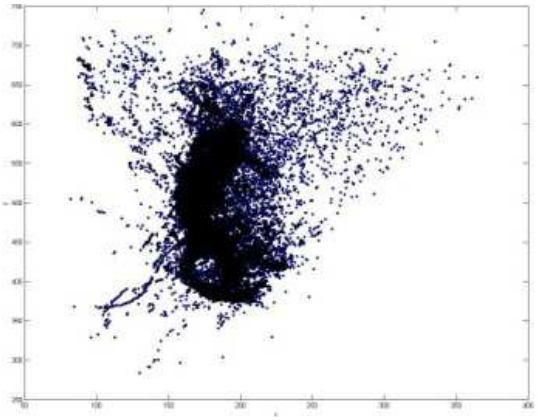

Figure 5: Locations visited by the operator (3 cameras)

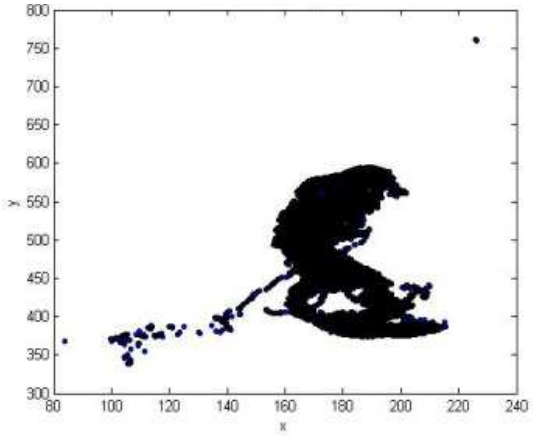

Figure 6: Locations visited by the operator (4 cameras)

Routing diagram

To see how the operator moves through the workstation, a routing diagram is constructed. To overcome the inaccuracies in the data, we take the moving average position over 1 second. In the routing diagram, these average locations are plotted for every 0.5 seconds of a total work cycle of 43.5 seconds. This results in a routing diagram as shown in Figure 7.

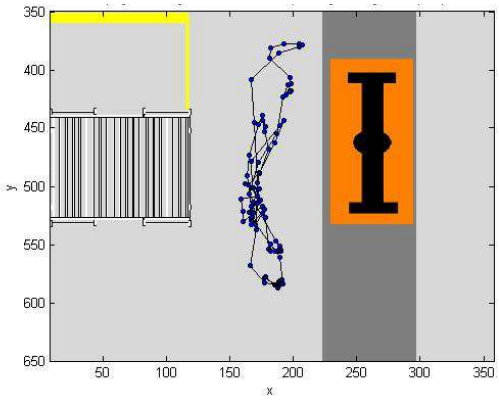

Figure 7: Routing diagram

Work load variation and fluctuation

Substantial variation in the cycle time can indicate the presence of high complexity in the work content. To investigate the possibility of deriving the cycle time and the fluctuation of this cycle time, a twofold method was used.
In the first phase, we select a (great) number of frames in the recordings and look for frames where the operator visits the same location he visits in the starting frame. The starting frames are randomly selected from the first minute of the video footage. To decrease the calculation time, we only calculate the distance of the operators starting position for a certain number of frames. Therefore, another random number is determined for every starting location and the method jumps through the data using this number as a search interval. For these selected frames, the method calculates the distance of the position in that frame to the starting position. If this distance is less than 3 cm, the frame number is saved in a list together with its corresponding starting frame. This list is used as the input for the second part of the method.

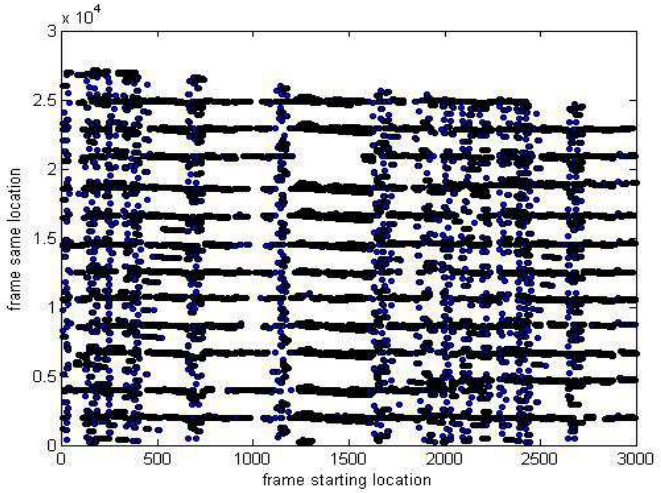

Figure 8: Samelocation Algorithm

Figure 8 shows the results of this method. On the x-axis, the selected starting frames are shown. The frames in which the operator visits the same location again, are shown on the y-axis. From a quick look at this graph, we learn that there is a clear pattern arising in these results.
The cycle time and its variation are calculated in the second phase of the method, by determining the distance between the lines in the graph above. For this, a 1-dimensional k-means clustering algorithm was implemented. This algorithm starts by determining the number of clusters in the dataset manually. We can derive from the graph that there are 12 clusters in this set. The starting solution can be constructed by selecting 12 randomly chosen points in the data range. Since we know that in the final solution the centroids of the clusters will be more or less evenly distributed over the data range, we divided the data range in 12 intervals and chose a random point in every interval. That way we were able to speed up the clustering algorithm.
The algorithm then calculates the distance of all data points to the centroids and assigns every point in the data set to the cluster with the closest centroid. Afterwards the new centroids of the clusters are calculated. The algorithm will repeat these steps until convergence.

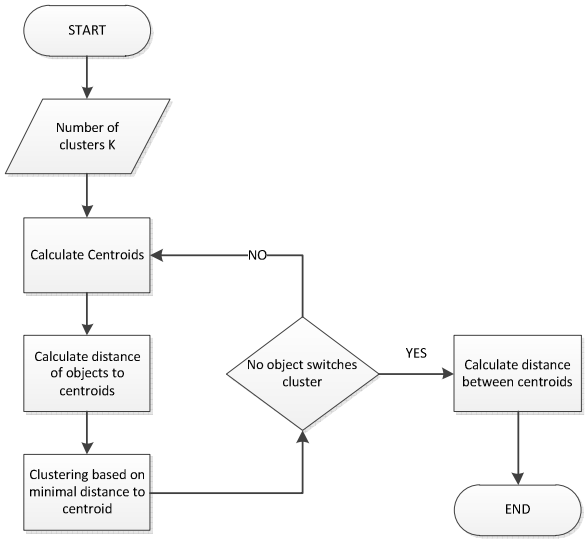

Figure 9: K-means clustering algorithm outline

The results of the clustering algorithm are shown in Table 1. The calculated cycle time of 43.5 seconds corresponds very well to the theoretical cycle time of 43.4 seconds the company take into account. Also the significant fluctuation in the work cycle agrees well with the video images.

Cycle time analysis		
Average C/T	43,5	sec
Standard Deviation	7,9	sec
Max/Min Ratios	57,42	%
	27,05	%

Table 1: Results cycle time analysis

Vehicle zones skipped

Unnecessary large walking distances may be an indication of a poorly designed process. To measure this, the area is divided in zones. Counting the number of times the operator passes such a zone without performing a value adding activity, can be useful to calculate the complexity and evaluate the design of the assembly process.
If we take a quick look at the video, we see that the operator passes the middle section of the axle quite often because a lot of the work is being done at both sides of the axle. The work station is divided in 4 zones, as shown in Figure 10.

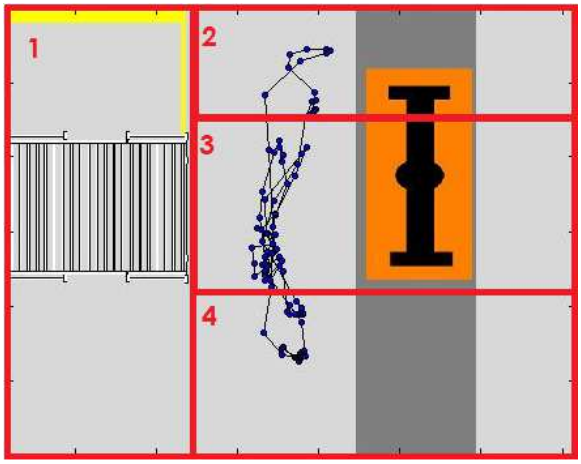

Figure 10: Vehicle zones skipped

To calculate this, an algorithm was implemented in matlab. The algorithm starts by determining in which zone the operator is located for every point in the data set. The pseudocode for this method is written down below. The input for this method is a consecutive list of the x- and y-coordinates of the locations visited by the operator and a list "zones" that is constructed as follows: [$x_{in,zone1}$, $x_{out,zone1}$, $y_{in,zone1}$, $y_{out,zone1}$, $x_{in,zone2}$,…].

function [inout] = ZoneDetect(path, zones)

INITIALIZE table "inout"

CALCULATE a,b and c coefficients for zone border equations

FOR length(path)
fill in the points in the equations of the zone borders to determine the position of these points to the borders

frame in	frame out	distance	time in	time out	theoretical walking time	real time	zone	1 if zone is skipped
16963	17167	2.78	339.74	343.82	0.05	4.08	4	0
17168	17245	19.29	343.84	345.38	0.36	1.54	**3**	**1**
17246	17674	30.11	345.40	353.96	0.56	8.56	2	0
17675	17730	8.29	353.98	355.08	0.15	1.10	**3**	**1**
17731	18239	6.65	355.10	365.26	0.12	10.16	4	0
18240	19026	13.18	365.28	381.00	0.24	15.72	3	0
19027	19215	0.10	381.02	384.78	0.00	3.76	4	0
19216	19305	19.29	384.80	386.58	0.36	1.78	**3**	**1**
19306	19671	33.84	386.60	393.90	0.63	7.30	2	0
19672	19728	10.99	393.92	395.04	0.20	1.12	**3**	**1**
19729	20229	5.93	395.06	405.06	0.11	10.00	4	0
20230	20644	10.46	405.08	413.36	0.19	8.28	3	0
20645	20705	5.24	413.38	414.58	0.10	1.20	4	0

Table 2: results vehicle zones skipped algorithm

```
FOR number of zones
        IF point is in zone
                        Add to inout
                        [X-coordinate, Y-coordinate, zone
                        number]
                BREAK;
                ENDIF
        ENDFOR
ENDFOR
```

Note that this method is not limited to 4 zones nor to rectangular zones.

To calculate this, the locations where the operator enters and leaves the zone are determined and the distance between these two points is calculated. Based on an average walking speed of 5km/h, the theoretical time the operator needs to cross the zone, is calculated. We can safely assume that the operator will perform some action in a zone, if he stays in that zone for a time that is significantly longer than that theoretical time.

We noticed that the real time needed to cross a zone is higher than the theoretical time based on an average walking speed of 5km/h. This can be explained by the fact that the operator usually isn't able to walk in a straight line and because he constantly needs to accelerate and decelerate. Therefore we say that the operator skips a zone if:

$$theoretical\ walking\ time.threshhold > real\ time\ in\ zone$$

An extract from these results (2 work cycles) is presented in Table 2. Again these results correspond well to the video, where we see that the operator regularly passes the middle section of the axle to go from the right hand side to the left and back.

CONCLUSIONS

In this paper, we presented an industrial proof-of-concept in which we investigated the possibility of automating the complexity analysis of an assembly workstation by using multi-camera video images. The current image processing technology can help us to automate the complexity analysis of a workstation of an assembly line.

For now, the focus was mainly on the position of the operator throughout his work cycle. More research should still be done on the recognition of hand motions and the viewing direction of the operator. Also linking the information we get from the video analysis with other sources of information such as MES-systems, ERP, RIFD tracking,… could be useful to translate more of the visual cues to complexity parameters.

REFERENCES

J.P. MacDuffie, K. Sethuraman, M.L. Fisher, "Product Variety and Manufacturing Performance: Evidence from the International Automotive Assembly Plant Study", Management Science, 1996, vol. 42 no. 3, pp. 350-369.

W.H. EIMaraghy, R.J. Urbanic, "Modelling of Manufacturing Systems Complexity", CIRP Annals, 2003, vol. 52, issue 1, pp.363-366.

W.H. EIMaraghy, R.J. Urbanic, "Assessment of Manufacturing Operational Complexity", CIRP Annals, 2004, vol. 53, issue 1, pp. 401–406.

L. Zeltzer, V. Limère, E.H. Aghezzaf, H. Van Landeghem, "Measuring the Objective Complexity of Assembly Workstations", In *Proceedings of the seventh international conference on computing in the global information technology (2012)*

D.W. Karger, W.M. Hancock, Advanced work measurement, Industrial press, N.Y. (1982)

Konz, S., Methods engineering. In Handbook of Industrial Engineering, edited by G. Salvendy, 3rd ed., pp. 1353–1390, 2001 (Wiley: New York).

M. Elnekave, I. Gilad, "Rapid video-based analysis system for advanced work measurement", In *International Journal of Production Research*, 2006, vol. 44, issue 2, pp. 271-290

B. Dencker, H-J. Balzer, W.E. Theuerkauf, M. Schweres, "Using a production-integrated video learning system (PVL) in the assembly sector of the car manufacturing industry", In *International Journal of Production Ergonomics*, 1999, Vol. 23, Issues 5-6, pp. 525-537

P. Taylor, "From figure skaters to the factory floor", 2011, *Financial Times*,http://www.ft.com/intl/cms/s/0/fc571624-ce98-11e0-a22c-00144feabdc0.html#axzz2Fbnaz0rp

A. Laurentini. 1994. The Visual Hull Concept for Silhouette-Based Image Understanding. *IEEE Trans. Pattern Anal. Mach. Intell*. 16, 2 (February 1994), 150-162.

Hansung Kim, Ryuuki Sakamoto, Itaru Kitahara, Tomoji Toriyama, and Kiyoshi Kogure.Robust foreground extraction technique using gaussian family model and multiple thresholds. In *Yasushi Yagi, Sing Bing Kang, In-So Kweon, and Hongbin Zha, editors, ACCV (1)*, volume 4843 of Lecture Notes in Computer Science, pages 758–768. Springer, 2007.

Zoran Zivkovic and Ferdinand van der Heijden. 2006. Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recogn. Lett. 27, 7 (May 2006), 773-